

d'après une idée du livre de Vincent MAILLE, complété par la lecture de *The Algorithm Beauty of Plants*, Przemyslaw PRUSINKIEWICZ et Aristid LINDENMAYER (page 7 et suivantes) <http://algorithmicbotany.org/papers/abop/abop.pdf> (attention ≈ 16 Mio)

1. Flocon de Van Koch

Soient les conventions

- F pour `forward(<n>)` (avance de $\langle n \rangle$ pas)
- *f* pour avancer sans tracer!
- L pour `left(60)` (tourne à gauche de 60°)
- R pour `right(60)` (tourne à droite de 60°)

Appliquées à un côté du flocon de Van Koch, cela signifie que chaque segment généré par F doit être transformé remplacé par le motif FLFRRFLF

Soit *w* la chaîne représentant la forme géométrique initiale, ici c'est un triangle équilatéral (par défaut la tortue Python est orientée à l'est); et *p* la chaîne représentant le motif de transformation.

```
1  # -*- coding:utf8 -*-
2  # python 3
3  #
4  # flocon de Van Koch et L-system
5
6  import turtle as tl
7
8  w, p = "FLLFLLFLL", "FLFRRFLF"
9  for n in range(3):
10     w=w.replace("F",p)
11  tl.penup() # pour déplacer le point de départ
12  tl.setpos(-100,20)
13  tl.pendown()
```



```
14 for c in w:
15     if c=="F":
16         tl.forward(10)
17     elif c=="L":
18         tl.left(60)
19     else:
20         tl.left(-60)
21 tl.mainloop()
```

2. À faire

modifier le programme afin d'obtenir d'autres fractales (changer la forme de départ, changer le motif de transformation, essayer avec des f , écrire des fonctions...)

Essayer d'importer le fichier comme un module afin d'avoir accès à une unique fonction pour faciliter les expériences.

En classe : rechercher la limite de l'aire du domaine?

```

1  # -*- coding:utf-8 -*-
2  # python3
3  #
4  # fractales et L-system
5
6  from turtle import *
7
8  def creeL(word,pattern,n):
9      """ transforme le motif initial <w> à l'aide de
10     ↪ <p>
11     <n> fois de suite
12     """
13     for i in range(n):
14         word=word.replace("F",pattern)
15     return(word)
16
17 def dessineL(liste,pas,angle):
18     """ liste la liste, F = forward(pas), L = left(
19     ↪ angle),
20     R = right(angle), f = forward(pas) sans tracer.
21     """
22     for c in liste:
23         if c=="F":
24             forward(pas)
25         elif c=="L":
26             left(angle)
27         elif c=="R":
28             right(angle)
29         else:
30             penup()
31             forward(pas)
32             pendown()

```



```
31
32 def maFractale(fig,motif,n,pas=10,angle=60):
33     """ dessine la fractale initialisée par la
    ↪ figure <fig>,
34     dans le quel <n> fois de suite "F" est remplacé
    ↪ par <motif>.
    """
35
36     penup()
37     setposition(-100,20)
38     pendown()
39     L=creel(fig,motif,n)
40     dessineL(L,pas,angle)
41     mainloop()
42
43 # ce qui suit n'est pas effectué si le fichier est
    ↪ chargé en
44 # tant que module, mais c'est ce qui est fait si on
    ↪ exécute le
45 # programme :
46 if __name__ == "__main__":
47     w, p, n = "FRFRFRFR", "FLFRFLF", 3
48     maFractale(w,p,n)
```
